STIC Search Report
EIC 2100

STIC Database Tracking Number: 174558

TO: Fred Ehichoya **Location: rnd 3b31** **Art Unit: 2162** **Monday, December 19, 2005** **Case Serial Number: 09/863638**	**From: Emory Damron** **Location: EIC 2100** **RND 4B19** **Phone: 571-272-3520** **Emory.Damron@uspto.gov**

Search Notes

Dear Fred,

Please find your fast and focused search results.
References of potential pertinence have been tagged, but please review all the packets in case you like something I didn't.
Of those references which have been tagged, please note any manual highlighting which I've done within the document.
In addition to searching on Dialog, I also searched EPO/JPO/Derwent.
There may be a few decent references contained herein, but I'll let you determine how useful they may be to you.
Please contact me if I can refocus or expand any aspect of this case, and please take a moment to provide any feedback (on the form provided) so EIC 2100 may better serve your needs. Good Luck!

Sincerely,
Emory Damron
Technical Information Specialist
EIC 2100, US Patent & Trademark Office
Phone: (571) 272-3520
Emory.damron@uspto.gov

16

STIC EIC 2100 174558
Search Request Form

Today's Date: **What date would you like to use to limit the search?**

Priority Date: 5/23/2001 Other:

Name FRED EHICHIOYA

AU 2162 Examiner # 79719

Room # RAN 3B31 Phone 2-4034

Serial # 09/863,538

Format for Search Results (Circle One):

PAPER DISK EMAIL

Where have you searched so far?

USP DWPI EPO JPO ACM IBM TDB

IEEE INSPEC SPI Other _____

Is this a "Fast & Focused" Search Request? (Circle One) YES NO

A "Fast & Focused" Search is completed in 2-3 hours (maximum). The search must be on a very specific topic and meet certain criteria. The criteria are posted in EIC2100 and on the EIC2100 NPL Web Page at http://ptoweb/patents/stic/stic-tc2100.htm.

What is the topic, novelty, motivation, utility, or other specific details defining the desired focus of this search? Please include the concepts, synonyms, keywords, acronyms, definitions, strategies, and anything else that helps to describe the topic. Please attach a copy of the abstract, background, brief summary, pertinent claims and any citations of relevant art you have found.

storing rows of a first table in a first storage module

storing rows of a second table in a second storage module

receiving a request to perform a join of first and second tab

distributing in response to the join request rows and one or more columns of the rows of the first table from the first storage module to the second storage module.

NCR

XCopy

STIC Searcher Linda Dana Phone 2-3520

Date picked up 12/19/05 Date Completed 12/19/05

sira Search and Information Resources Administration

```
Set     Items   Description
S1     483430   COMBINE? OR COMBINING? OR MERGE? OR MERGING? OR AMALGAM?
S2     188902   BLEND? OR (ADD OR ADDS OR ADDING OR ADDED OR ADDITION)()TO-
              GETHER?
S3      11453   COMMINGL? OR CONJOIN? OR MINGL? OR INTERMIX? OR INTERMINGL?
              OR COMMIX? OR BIND?()TOGETHER?
S4     784805   FUSE? OR FUSING? OR MELD? OR UNITE? OR UNITING? OR INTERBL-
              END? OR INTEGRAT?
S5     871967   INTERWEAV? OR INTERWOVEN? OR HOMOG? OR JOIN?
S6    3853183   FIRST? OR 1ST OR PRIMARY OR INITIAL? OR ORIGINAL? OR LEADO-
              FF? OR MAIN OR CHIEF OR INTRODUCTORY?
S7    3961864   SECOND? OR 2ND OR DOUBL? OR TWIN? OR EXTRA? OR DUPLICAT? OR
              ANOTHER OR SUBSIDIAR? OR AUXILIAR?
S8       2676   STORAG?()MODUL?
S9      29019   VOLATIL?(3N)(MEMOR? OR STORAG? OR CACHE? OR BUFFER?)
S10    206557   RAM? ? OR SRAM? ? OR DRAM? ? OR SDRAM? ? OR VRAM? ? OR RAN-
              DOM()ACCESS()MEMOR? OR CDROM? OR CD()ROM? ?
S11    156841   ROM? ? OR PROM? ? OR EPROM? ? OR EEPROM? ? OR NONVOLATIL? -
              OR NON()VOLATIL? OR NVROM? OR READ()ONLY()MEMOR?
S12     42107   HD OR HDS OR HARDDRIVE? OR HARDDISC? OR HARDDISK? OR HARD(-
              )(DRIVE? OR DISC? OR DISK?) OR DISCDRIVE? OR DISKDRIVE?
S13     70690   (DISC OR DISK)()DRIVE? OR STOR?()(DISK? OR DISC?) OR HDD OR
              HDDS OR CDRW OR CDW OR CD()(W OR RW)
S14    180001   ROW? ?
S15    394260   TABLE? ?
S16    243251   COLUMN? ?
S17   2298205   PLURAL? OR MULTIP? OR MULTIT? OR SEVERAL? OR NUMEROUS? OR -
              MANY
S18   1270169   IC=G06F?
S19    967055   MC=T01?
S20         9   S1:S5 AND S8:S13 AND S14 AND S15 AND S16
S21         6   S20 AND (S18:S19 OR S6:S7)
S22       324   S6(5N)S14:S16 AND S7(5N)S14:S16 AND S17 AND S8:S13
S23        57   S22 AND S1:S5
S24        97   S22 AND S18:S19
S25       145   S20:S21 OR S23:S24
S26    832282   PR=2002:2005
S27       143   S25 NOT S26
S28       143   IDPAT (sorted in duplicate/non-duplicate order)
File 347:JAPIO Nov 1976-2005/Jul(Updated 051102)
         (c) 2005 JPO & JAPIO
File 350:Derwent WPIX 1963-2005/UD,UM &UP=200581
         (c) 2005  Thomson Derwent
```

28/3,K/7 (Item 7 from file: 350)
DIALOG(R)File 350:Derwent WPIX

016606423 **Image available**
WPI Acc No: 2004-765157/200475
Related WPI Acc No: 2004-477206; 2005-345213
XRPX Acc No: N04-603619

Database search result provision system for e.g. personal computer, invokes spooling module when relational procedure which performs operation on rows that are copied into memory location, of application module, fails

Patent Assignee: MICROSOFT CORP (MICT)
Inventor: BUNKER R T; GALINDO-LEGARIA C A; GRAEFE G; JOSHI M M
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 20040205078	A1	20041014	US 2001800379	A	20010306	200475	B
			US 2004829595	A	20040422		

Priority Applications (No Type Date): US 2001800379 A 20010306; US 2004829595 A 20040422
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20040205078	A1		22	G06F-007/00	Cont of application US 2001800379 Cont of patent US 6748392

... **for e.g. personal computer, invokes spooling module when relational procedure which performs operation on** rows **that are copied into memory location, of application module, fails**

Abstract (Basic):

... A spool module receives several set of **rows** and **columns** and copies the **rows** that have common values for **columns** , with respect to specific relation, in predetermined memory location. Application module invokes relational procedure to perform specific operation on copied **rows** and to output the processed results. Spooling module is invoked when the relational procedure is...

... 1) a method of evaluating semi- **join** ;
(...

...2) recorded medium storing semi- **join** evaluating program...

...3) a method of evaluating an anti-semi- **join** ;
(...

...4) a recorded medium storing anti-semi- **join** evaluating program...

...5) a method of sorting **rows** of database **table** ;
(...

...6) recorded medium storing program for **row** sorting...

...unnecessary spooling of an entire sub-expression result. Avoids the need of storing entire operand **table** in memory at once and thereby decreasing time and resources necessary to evaluate a **join** . Avoids the need of physical copy of each **row** in spool by buffering the **rows** of the input **table** , thereby optimizing the stored **rows** and achieving higher efficiency...

...magnetic **disk drive** (151...

...optical **disk** **drive** (155
...Title Terms: **ROW** ;
International Patent Class (Main): **G06F-007/00**
Manual Codes (EPI/S-X): **T01-J05B4M** ...

... **T01-S03**

US 20040205078A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2004/0205078 A1

Galindo-Legaria et al. (43) Pub. Date: Oct. 14, 2004

(54) SYSTEM AND METHOD FOR SEGMENTED EVALUATION OF DATABASE QUERIES

(75) Inventors: Cesar A. Galindo-Legaria, Redmond, WA (US); Goetz Graefe, Bellevue, WA (US); Milind M. Joshi, Bellevue, WA (US); Ross T. Bunker, Seattle, WA (US)

Correspondence Address:
WOODCOCK WASHBURN LLP
ONE LIBERTY PLACE, 46TH FLOOR
1650 MARKET STREET
PHILADELPHIA, PA 19103 (US)

(73) Assignee: Microsoft Corporation, Redmond, WA (US)

(21) Appl. No.: 10/829,595

(22) Filed: Apr. 22, 2004

Related U.S. Application Data

(63) Continuation of application No. 09/800,379, filed on Mar. 6, 2001, now Pat. No. 6,748,392.

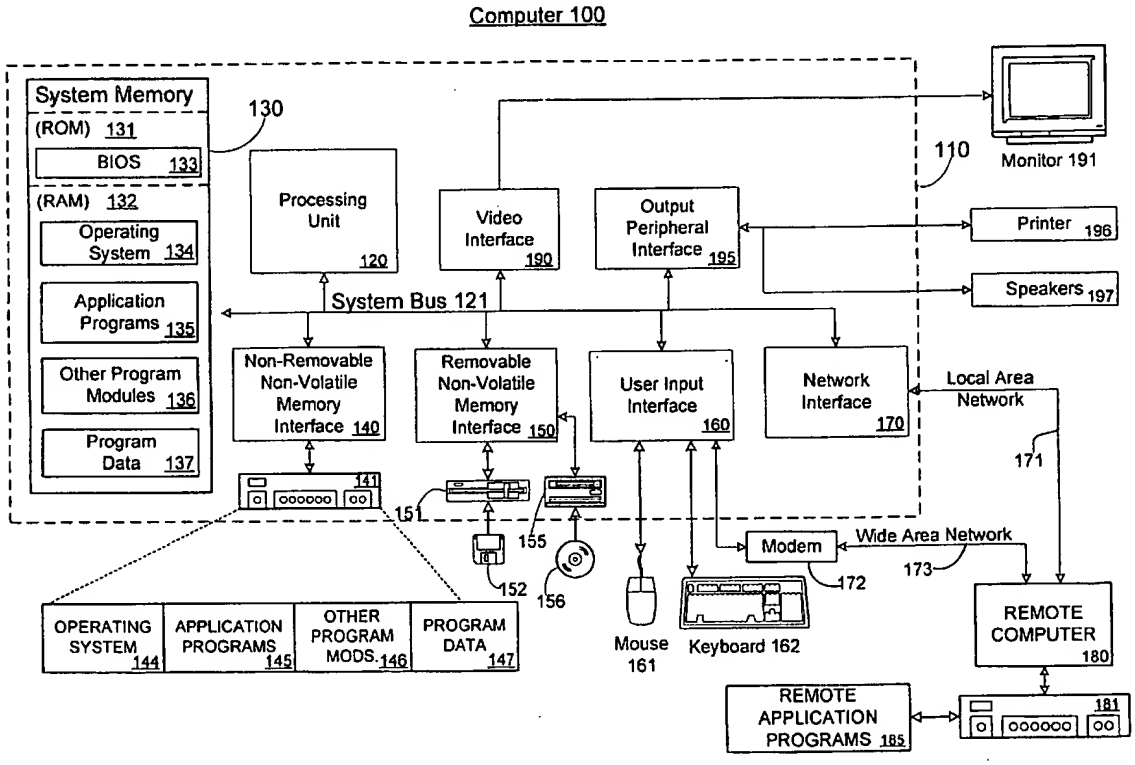
Publication Classification

(51) Int. Cl.[7] G06F 7/00

(52) U.S. Cl. 707/100

(57) ABSTRACT

A method of satisfying a database query includes evaluating certain joins on a per-segment basis. An expression tree is produced for the query, and the expression tree is evaluated to identify joins whose operands are two instances of the same relation and whose join predicate conjunctively includes an equality comparison between two instances of the same column. When such a join is identified, it may be evaluated by segmenting the operand relation according to the columns that are compared for equality in the predicate. The join is then evaluated by performing the join operation separately on each segment. Segments may be spooled separately, thereby exploiting the efficiencies obtained by spooling even where the entire relation is too large to fit in the spool. Execution iterators are provided for spooling successive segments and for applying the join to the spooled segment.



PGPUB-DOCUMENT-NUMBER: **20040205078**

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20040205078 A1

TITLE: System and method for segmented evaluation of database queries

PUBLICATION-DATE: October 14, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Galindo-Legaria, Cesar A.	Redmond	WA	US	
Graefe, Goetz	Bellevue	WA	US	
Joshi, Milind M.	Bellevue	WA	US	
Bunker, Ross T.	Seattle	WA	US	

US-CL-CURRENT: 707/100

CLAIMS:

1-33. (canceled)

34. A system for producing results from a database comprising: a spooling module which receives a relation comprising a set of **rows** and a set of **columns**, and which copies into a memory location one or more **rows** of said relation having common values for one or more **columns** of said relation; and an application module which invokes a relational procedure, wherein said relational procedure performs a relational operation on **rows** in said memory location to produce one or more result **rows**, said application module invoking said spooling module when said relational procedure is unable to produce result **rows**.

35. The system of claim 34, further comprising: a query compiler which receives a query and produces an expression tree based on said query, said relational procedure being based on said expression tree.

36. The system of claim 35, wherein said expression tree comprises a **join** operator having: a first operand which includes a first instance of a relation, said relation having a set of **rows** and a set of **columns**; a second operand including a second instance of said relation; and a first predicate; said system further comprising: an optimization module which receives said expression tree and determines that said predicate is, or conjunctively includes, an equality comparison between one or more **columns** of said first instance of said relation and corresponding one or more **columns** of said second instance of said relation, said one or more **columns** of said first and second instances of said relation being the same **columns** as said one or more **columns** of said relation.

37. The system of claim 34, wherein said relational procedure comprises a **join**.

38. The system of claim 37, wherein said **join** comprises one of: inner **join**, semijoin, or anti-semijoin.

39. A method of evaluating a semijoin having a first operand, a second operand, and a predicate, said method comprising the acts of: determining that said first and second operands each comprise first and second instances, respectively, of a common relation; determining that said predicate is, or conjunctively includes, an equality comparison between one or more **columns** of said first instance of said relation and corresponding **columns** of said second instance of said relation; segmenting said common relation based on said one or more **columns** to produce one or more segments of said common relation; and performing said semijoin separately on each of said segments.

40. The method of claim 39, further comprising the act of spooling each of

said segments in a memory location, wherein said performing act comprises
applying said semijoin successively to the segments spooled in said memory
location.

41. The method of claim 39, further comprising the act of compiling a SQL
query which includes an EXISTS clause to produce a relational expression that
includes said semijoin.

42. A computer-readable medium having computer-executable instructions to
perform the method of claim 39.

43. A method of evaluating an anti-semijoin having a first operand, a second
operand, and a predicate, said method comprising the acts of: determining that
said first and second operands each comprise first and second instances,
respectively, of a common relation; determining that said predicate is, or
conjunctively includes, an equality comparison between one or more **columns** of
said first instance of said relation and corresponding **columns** of said second
instance of said relation; segmenting said common relation based on said one
or more **columns** to produce one or more segments of said common relation; and
performing said anti-semijoin separately on each of said segments.

44. The method of claim 43, further comprising the act of spooling each of
said segments in a memory location, wherein said performing act comprises
applying said anti-semijoin successively to the segments spooled in said memory
location.

45. The method of claim 43, further comprising the act of compiling a SQL
query which includes a NOT EXISTS clause to produce a relational expression
that includes said anti-semijoin.

46. A computer-readable medium having computer-executable instructions to

perform the method of claim 43.

47. A method of sorting **rows** of a database **table** according to a first set of one or more **columns, said rows** having been sorted on a second set of one or more **columns,** said method comprising the acts of: segmenting said database **table** based on said second set of **columns** to produce segments of said database **table,** each of said segments comprising **rows** having common values in said second set of **columns**; and separately sorting each of said segments based on the values in said first set of **columns**.

48. The method of claim 47, wherein said segmenting act is performed without spooling **rows** of said segments.

49. The method of claim 47, wherein said segmenting act comprises: identifying a first **row** of said database **table,** said first **rows** having first values in said second set of **columns**; and identifying a second **row** of said database **table,** said second **row** having said first values in said second set of **columns,** said database **table** having an order, said second **row** being the last occurring **row** in said order having said first values in said second set of **columns**.

50. The method of claim 47, wherein said act of separately sorting comprises: invoking a first function which sorts **rows** in first one of said segments; invoking a second function which identifies a next one of said segments; and repeating said acts of invoking said first and second function until all of said segments have been exhausted.

51. The method of claim 47, further comprising the act of: creating an expression tree including a GbApply operator having: a first child sub-tree specifying said database **table** as input to said GbApply operator; segmentation data indicating that said database **table** is to be segmented on said second set

of **columns**; and a second child sub-tree specifying a relational fragment to be performed on each of said segments, said relational fragment specifying a sort of the **rows** of said segments on said first set of **columns**.

52. A computer-readable medium having computer-executable instructions to perform the method of claim 47.

53. In a database system which performs operations on a **table having rows and columns**, a method of identifying a **row** having a superlative value in a first of said **columns** from among a set of **rows** having a common value in a second of said **columns**, said method comprising the acts of: segmenting said **rows** based on the values in said second **column** to produce groups of one or more **rows, wherein all of the rows** in a first of said groups have a common value in said second **column**; sorting the **rows** in said first group based on the values in said first **column**; and identifying the first or last **row** in said first group.

54. The method of claim 53, wherein said superlative value comprises a maximum value, and wherein said identifying act comprises identifying the last **row** in said first group.

55. The method of claim 53, wherein said superlative value comprises a minimum value, and wherein said identifying act comprises identifying the first **row** in said first group.

56. The method of claim 53, wherein said segmenting act comprises: sorting said **rows** based on the values in said second **column**; and following said sorting act, identifying the first and last **rows** which have said common value in said second **column**.

57. The method of claim 53, further comprising the act of: creating an expression tree including a GbApply operator having: a first child sub-tree

specifying said **table** as input to said GbApply operator; segmentation data indicating that said database **table** is to be segmented on said second set of **columns**; and a second child sub-tree specifying a relational fragment to be performed on each of said segments, said relational fragment specifying the selection of either the first or last of the **rows** in a segment.

58. A computer-readable medium having computer-executable instructions to perform the method of claim 53.

59. A system for evaluating a database query comprising: a query compiler which receives the query and produces an expression tree based on the query; an analysis module which identifies an expression tree having a **join** operator which includes: a first operand which includes a first instance of a relation based on information stored in said database, said relation having a set of **rows** and a set of **columns**; a second operand including a second instance of said relation; and a first predicate which is, or conjunctively includes, an equality comparison between one or more **columns** of said first instance of said relation and corresponding one or more **columns** of said second instance of said relation; and an evaluation module which segments said relation according to the values in said one or more **columns** and which applies said **join** operator separately to each segment of said relation.

60. The system of claim 59, further comprising: a spooling module which spools **rows** of each segment in a designated memory location; wherein said evaluation module applies said **join** operator to the **rows** stored in said designated memory location.

61. The system of claim 59, wherein said evaluation module does not spool the segments of said relation.

62. The system of claim 59, wherein said **join** operator comprises one of: inner
join, semijoin, or anti-semijoin.

63. The system of claim 59, wherein said analysis module determines that said
first operand further includes a filter which modifies said first instance of
said relation according to a second predicate.

64. The system of claim 59, wherein said analysis module determines that said
first operand further includes an aggregate operation which specifies the
computation of a value based on one or more **rows** of said relation.

65. The system of claim 59, wherein said relation comprises a stored **table**.

66. The system of claim 59, wherein said relation comprises a sub-expression
based on one or more stored **tables**, and wherein said system further includes a
module which evaluates said sub-expression to produce said relation.

28/3,K/48 (Item 48 from file: 350)
DIALOG(R)File 350:Derwent WPIX

010421383 **Image available**
WPI Acc No: 1995-322699/199542
XRPX Acc No: N95-242922
Database join **processing system -** joins **relations based on** join **fields in relational database**
Patent Assignee: MITSUBISHI DENKI KK (MITQ); MITSUBISHI ELECTRIC CORP (MITQ)
Inventor: MATSUMOTO T
Number of Countries: 003 Number of Patents: 004
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
GB 2287807	A	19950927	GB 952768	A	19950213	199542	B
JP 7253991	A	19951003	JP 9445620	A	19940316	199548	
US 5613142	A	19970318	US 95388616	A	19950214	199717	
GB 2287807	B	19980506	GB 952768	A	19950213	199820	

Priority Applications (No Type Date): JP 9445620 A 19940316
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
GB 2287807	A		124	G06F-007/32	
JP 7253991	A		24	G06F-017/30	
US 5613142	A		48	G06F-007/08	
GB 2287807	B			G06F-007/32	

Database join **processing system**...

... joins **relations based on** join **fields in relational database**

...Abstract (Basic): The system **joins** distributed data with a **join** key and produces a **joined** table (100,200). Recording devices, **disk drives** (4a-4d) store the distributed data e.g. employee data (5a-5d) and sales data...

...data. The main processor (1) receives the data from the slave-processors and produces the **joined** table...

...Each slave-processor checks a **join** key of the second data, sales data, with a **join** key of the first data. Based on the checking result it selects the second data...

...USE/ADVANTAGE - Provides high speed **joining** , eliminates burden on master processor as data distributed and stored in slave processors, part of **join** processing done in parallel...
...Abstract (Equivalent): A **join** processing system which **joins** distributed first and second data, the first data including a first **join** key, the second data including a second **join** key and produces a **joined** table, the **join** processing system comprising...

...a) a **plurality** of recording means for storing the distributed first and second data...

...b) a **plurality** of sub-processor means, each respectively coupled to a respective recording means of the **plurality** of recording means, for retrieving the first and second data from the respective recording means...

...c) main processor means for receiving the first and second data from the

plurality of sub-processor means and producing the **joined** table...

...wherein each of the sub-processor means retrieves the first data with the first **join** key from the respective recording means and produces a **first** sub- **table** including the **first** **join** key based on the first data including a predetermined value, and transfers the first subtable ...

...wherein the main processor means, receives the **first** sub- **tables** from the **plurality** of sub-processor means, and produces a **first** **main** **table** ;
...

...sub-processor means retrieves the second data in the respective recording means with a second **join** key equal to a first **join** key included in the **first** main **table** , produces a **second** sub- **table** based on the retrieved **second** data, and transfers the **second** sub- **table** to the **main** processor means; and...

...wherein the main processor means receives the **second** sub- **table** from the **plurality** of sub-processor means, produces a **second** **main** **table** , **joins** the **first** and **second** data based on the **first** and **second** **main** **tables** , and produces the **joined** **table** .

...Title Terms: **JOIN** ;
International Patent Class (Main): **G06F-007/08** ...

... **G06F-007/32** ...

... **G06F-017/30**
International Patent Class (Additional): **G06F-007/14** ...

... **G06F-007/36** ...

... **G06F-012/00** ...

... **G06F-012/04**
Manual Codes (EPI/S-X): **T01-F05A** ...

... **T01-J05B4**

US005613142A

United States Patent [19]

Matsumoto

[11] Patent Number: 5,613,142

[45] Date of Patent: Mar. 18, 1997

[54] **JOIN PROCESSING SYSTEM AND METHOD FOR JOINING RELATIONS BASED ON JOIN FIELDS IN A RELATIONAL DATABASE**

[75] Inventor: **Toshio Matsumoto**, Kanagawa, Japan

[73] Assignee: **Mitsubishi Denki Kabushiki Kaisha**, Tokyo, Japan

[21] Appl. No.: **388,616**

[22] Filed: **Feb. 14, 1995**

[30] **Foreign Application Priority Data**

Mar. 16, 1994 [JP] Japan 6-045620

[51] **Int. Cl.**[6] **G06F 7/08**; G06F 7/14; G06F 7/36; G06F 12/04

[52] **U.S. Cl.** **395/602**; 364/DIG. 1; 364/DIG. 2

[58] **Field of Search** ..... 395/800, 600, 395/160, 200.07, 200.09, 650, 700; 364/DIG. 1, DIG. 2

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,774,657 9/1988 Anderson et al. 395/600
5,193,182 3/1993 Bachman et al. 395/600
5,210,870 5/1993 Baum et al. 395/600

FOREIGN PATENT DOCUMENTS

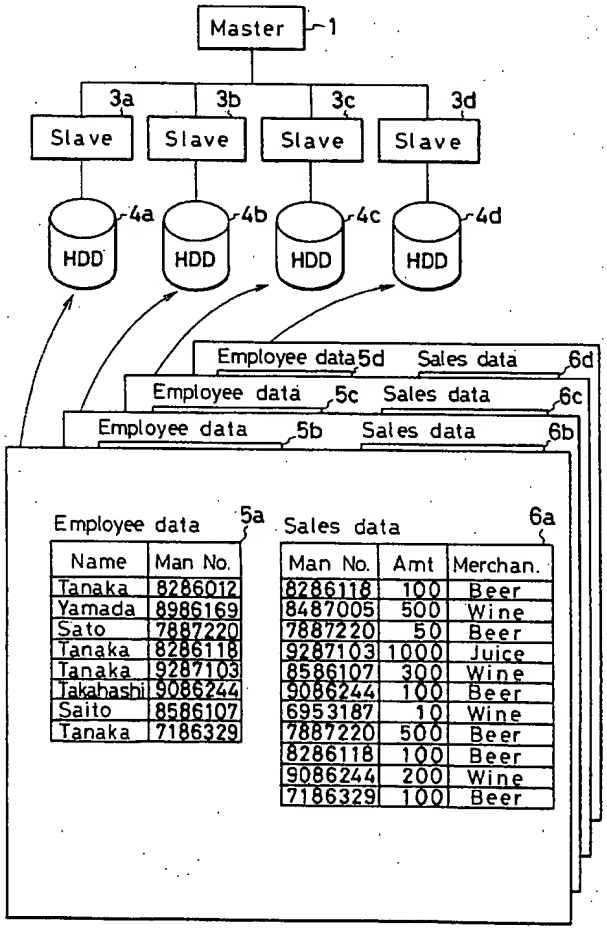
61-273633 12/1986 Japan .
3-156571 7/1991 Japan .

Primary Examiner—Daniel H. Pan
Attorney, Agent, or Firm—Wolf, Greenfield & Sacks, P.C.

[57] **ABSTRACT**

This invention provides a join processing system and method, which operates efficiently without a burden to a master processor, in a relational database on a multiprocessor. In a system which includes a master processor, multiple slave processors and disk drives, data is distributed and stored in the disk drives. Each of the slave processors retrieves a first sub-table and transfers the table to the master processor, and the master processor creates a first main table. Each of the slave processors retrieves the second data selectively with reference to the first main table, and creates a second sub-table. The master processor merges the second sub-tables, and creates a second main table. Then, a join processing of the first main table and the second main table is performed.

16 Claims, 23 Drawing Sheets



Name	Man No.
Tanaka	8286012
Yamada	8986169
Sato	7887220
Tanaka	8286118
Tanaka	9287103
Takahashi	9086244
Saito	8586107
Tanaka	7186329

Man No.	Amt	Merchan.
8286118	100	Beer
8487005	500	Wine
7887220	50	Beer
9287103	1000	Juice
8586107	300	Wine
9086244	100	Beer
6953187	10	Wine
7887220	500	Beer
8286118	100	Beer
9086244	200	Wine
7186329	100	Beer

In embodiment 8, a burden to the master processor is further lessened relative to embodiment 7. Since the slave processors transfer the data among the slave processors, a burden to the master processor is further lessened.

Embodiment 9.

Thus far, operations such as transferring the tables, referring to the tables, transferring the hash tables, referring to the hash tables, etc. have been explained. The operations of referring to the tables and charts may be replaced with the operations of transferring the tables and charts. The operations of transferring the tables and charts may be replaced with the operations of referring to the tables and charts.

Regarding referring to the tables and charts, a traffic volume between the master processor and the slave processors is increased. However, the data exists either in the master processor or the slave processors, a merit exists that a memory field in another processor is not pressured. Meanwhile, in the operation of transferring data, a disadvantage exists that the memory in the processor, to which the data is transferred, is pressured. However, when the tables and charts are transferred once, the reference relationship between the master processor and the slave processors is diminished. Hence, the traffic volume between the master processor and the slave processors is able to be lessened.

Embodiment 10.

In the previously described embodiments, the operations in which the records of the employee data and the sales data are used directly and the operations in which the addresses of the employee data and the sales data are used are shown. However, these operations of using records and using addresses may be combined. For example, the man numbers and the addresses may be retrieved from the employee data, while the man numbers and the records, themselves, are retrieved from the sales data. Or, vice versa is possible.

Embodiment 11.

In embodiment 10, two kinds of data, the employee data and the sales data, are used. Alternatively, either three or four kinds of data may be joined in the join processing.

Having thus described one particular embodiment of the invention, various alterations, modifications, and improvements will readily occur to those skilled in the art. Such alterations, modifications, and improvements are intended to be part of this disclosure, and are intended to be within the spirit and scope of the invention. Accordingly, the foregoing description is by way of example only and is limited only as defined in the following claims and the equivalents thereto.

What is claimed is:

1. A join processing system which joins distributed first and second data, the first data including a first join key, the second data including a second join key and produces a joined table, the join processing system comprising:

(a) a plurality of recording means for storing the distributed first and second data;

(b) a plurality of sub-processor means, each respectively coupled to a respective recording means of the plurality of recording means, for retrieving the first and second data from the respective recording means and outputting the first and second data; and

(c) main processor means for receiving the first and second data from the plurality of sub-processor means and producing the joined table;

wherein each of the sub-processor means retrieves the first data with the first join key from the respective recording means and produces a first sub-table including the first join key based on the first data including a predetermined value, and transfers the first subtable to the main processor means;

wherein the main processor means, receives the first sub-tables from the plurality of sub-processor means, and produces a first main table;

wherein each of the sub-processor means retrieves the second data in the respective recording means with a second join key equal to a first join key included in the first main table, produces a second sub-table based on the retrieved second data, and transfers the second sub-table to the main processor means; and

wherein the main processor means receives the second sub-table from the plurality of sub-processor means, produces a second main table, joins the first and second data based on the first and second main tables, and produces the joined table.

2. A join processing system which joins distributed first and second data, the first data including a first join key, the second data including a second join key and produces a joined table, the join processing system comprising:

(a) a plurality of recording means for storing the distributed first and second data;

(b) a plurality of sub-processor means, each respectively coupled to a respective recording means of the plurality of recording means, for retrieving the first and second data from the respective recording means and outputting the first and second data; and

(c) main processor means for receiving the first and second data from the plurality of sub-processor means and producing the joined table;

wherein each of the sub-processor means retrieves the first data with the first join key from the respective recording means and produces a first sub-table including the first join key based on the retrieved first data including a predetermined value, and transfers the first sub-table to the main processor means;

wherein the main processor means receives the first sub-tables from the plurality of sub-processor means, produces a first main table, calculates a hash value of each first join key in the first main table, produces a hash table showing an existence of the hash values and transfers the hash table to each of the plurality of sub-processor means;

wherein each of the plurality of sub-processor means receives the hash table, calculates a second join key hash value of the second join key of the second data, checks an existence of the calculated second join key hash value in the hash table, selects a second data whose second join key hash value is confirmed to exist in the hash table, produces a second sub-table based on selected second data, and transfers the second sub-table to the main processor means; and

wherein the main processor means receives the second sub-table from each of the plurality of sub-processor means, produces a second main table, joins the first and second data based on the first and second main tables, and produces the joined table.

3. A join processing system which joins distributed first and second data, the first data including a first join key, the second data including a second join key and produces a joined table, the join processing system comprising:

(a) a plurality of recording means for storing the distributed first and second data;

(b) a plurality of sub-processor means, each respectively coupled to a respective recording means of the plurality

of recording means, for retrieving the first and second data from the respective recording means and outputting the first and second data; and

(c) main processor means for receiving the first and second data from the plurality of sub-processor means and producing the joined table;

wherein each of the plurality of sub-processor means extracts the first join key of the first data and a first address of the first data in the respective recording means and produces a first subtable based on the extracted first join key and the first address, and transfers the first sub-table to the main processor means;

wherein the main processor means receives the first sub-tables from the plurality of sub-processor means, and produces a first main table;

wherein each of the plurality of sub-processor means refers to the first main table, compares the second join key of the second data with the first join keys included in the first main table, selects second data whose second join key is matched with one of the first join keys in the first main table, produces a second sub-table based on the selected second data, and transfers the second sub-table to the main processor means; and

wherein the main processor means receives the second sub-table from the plurality of sub-processor means, produces a second main table, joins the first and second data based on the first and second main tables, and produces the joined table.

4. The join processing system of the claim **3**, wherein

the main processor means provides a flag field corresponding to each of the first data in the first main table,

wherein each of the plurality of sub-processor means stores a comparison result in the flag field when the subprocessor means compares the second join key of the second data with the first join keys included in the first main table to select the second data; and

wherein the main processor means joins the first and second data according to a comparison result stored in the flag field in the first main table.

5. A join processing system which joins distributed first and second data, the first data including a first join key, the second data including a second join key and produces a joined table, the join processing system comprising:

(a) a plurality of recording means for storing the distributed first and second data;

(b) a plurality of sub-processor means, each respectively coupled to a respective recording means of the plurality of recording means, for retrieving the first and second data from the respective recording means and outputting the first and second data; and

(c) main processor means for receiving the first and second data from the plurality of sub-processor means and producing the joined table;

wherein each of the plurality of sub-processor means retrieves the first data with the first join key from the respective recording means and produces a first subtable including the first join key based on the first data including a predetermined value, and transfers the first sub-table to the main processor means;

wherein the main processor means receives the first sub-tables from the plurality of sub-processor means, produces a first main table, calculates a hash value of each first join key in the first main table, produces a hash table showing an existence of the hash values, and transfers the hash table to each of the plurality of sub-processor means;

wherein each of the plurality of sub-processor means receives the hash table, calculates a second join key hash value of the second join key of the second data, checks an existence of the calculated second join key hash value in the hash table, extracts a second join key and an address of the second data whose second join key hash value is confirmed to exist in the hash table, produces a second sub-table based on the extracted second join key and address of the second data, and transfers the second sub-table to the main processor means; and

wherein the main processor means receives the second sub-table from the plurality of sub-processor means, produces a second main table, joins the first and second data based on the first and second main tables, and produces the joined table.

6. A join processing system which joins distributed first and second data, the first data including a first join key, the second data including a second join key and produces a joined table, the join processing system comprising:

(a) a plurality of recording means for storing the distributed first and second data;

(b) a plurality of sub-processor means, each respectively coupled to a respective recording means of the plurality of recording means, for retrieving the first and second data from the respective recording means and outputting the first and second data; and

(c) main processor means for receiving the first and second data from the plurality of sub-processor means and producing the joined table;

wherein each of the plurality of sub-processor means retrieves the first data with the first join key from the respective recording means and produces a first subtable including the first join key based on the first data including a predetermined value, and transfers the first sub-table to the main processor means;

wherein the main processor means receives the first sub-tables from the plurality of sub-processor means, produces a first main table, and transfers the first main table to each of the plurality of sub-processor means;

wherein each of the plurality of sub-processor means receives the first main table, retrieves the second data in the respective recording means with the second join key equal to a first join key included in the received first main table, joins the first data-in the received first main table and the retrieved second data based on the second join key, produces a joined sub-table, and transfers the joined sub-table to the main processor means; and

wherein the main processor means receives the joined sub-table from the plurality of sub-processor means, and produces the joined table.

7. The join processing system of claim **6**, wherein each of the plurality of sub-processor means extracts the second join key and an address of the second data stored in the respective recording means, compares the extracted second join key of the second data with the first join keys included in the first main table, extracts second data from the recording means based on the extracted address according to the comparison result, and produces the joined sub-table.

8. The join processing system of any one of claims **1**, **2**, **3**, **4**, or **5**, wherein each of the plurality of sub-processor means includes data transfer means for transferring data among the plurality of sub-processor means.

9. A join processing method for joining distributed first and second data, the first data including a first join key, the second data including a second join key and producing a

joined table, for a database system including a plurality of recording means for storing the distributed first and second data, a plurality of sub-processors, each coupled to the plurality of recording means respectively, and a main processor coupled to the plurality of sub-processors for controlling the plurality of sub-processors and producing the joined table, the join processing method comprising the steps of:

in each of the plurality of sub-processors, retrieving the first data with the first join key from the respective recording means, sorting the retrieved first data with the first join key so as to produce a first sub-table, and transferring the first sub-table to the main processor;

in the main processor, receiving the first sub-table from each of the plurality of sub-processors, merge-sorting the first sub-tables with the first join key, and producing a first main table;

in each of the plurality of sub-processors, retrieving the second data from the respective recording means, sorting the second data, extracting the second data, referring to the first join key included in the first main table, so as to produce a second sub-table, and transferring the second sub-table to the main processor; and

in the main processor, receiving the second subtables from each of the plurality of sub-processors, merge-sorting the second sub-tables with the second join key so as to produce a second main table, joining the first and second data based on the first and second main tables, and producing the joined table.

10. A join processing method, for joining distributed first and second data, the first data including a first join key, the second data including a second join key and producing a joined table, for a database system including a plurality of recording means for storing the distributed first and second data, a plurality of subprocessors each coupled to one of the plurality of recording means respectively, and a main processor coupled to the plurality of subprocessors for controlling the plurality of sub-processors and producing the joined table, the join processing method comprising the steps of:

in each of the plurality of sub-processors, retrieving the first data with the first join key from the respective recording means when the first data includes a predetermined value and sorting the retrieved first data with the first join key so as to produce a first sub-table, and transferring the first sub-table to the main processor;

in the main processor, receiving the first subtables from each of the plurality of sub-processors, merge sorting the first sub-tables with the first join key so as to produce a first main table, calculating first hash values of the first join keys in the first main table, producing a first hash table showing an existence of the first hash values, and transferring the first hash table to each of the plurality of sub-processors;

in each of the plurality of sub-processors, receiving the first hash table, calculating a second hash value of a second join key of the second data, checking an existence of the calculated second hash value in the first hash table, selecting second data whose calculated second hash value is confirmed to exist in the first hash table, sorting the selected second data so as to produce a second sub-table, and transferring the second sub-table to the main processor;

in the main processor, receiving the second subtables from each of the plurality of sub-processors, merge sorting the second sub-tables with the second join key so as to produce a second main table, joining the first and second data based on the first and second main tables, and producing the joined table.

11. A join processing method for joining distributed first and second data, the first data including a first join key, the second data including a second join key and producing a joined table, for a database system including a plurality of recording means for storing the distributed first and second data, a plurality of sub-processors each coupled to one of the plurality of recording means respectively, and a main processor coupled to the plurality of sub-processors for controlling the plurality of sub-processors and producing the joined table, the join processing method comprising the steps of:

in each of the plurality of sub-processors, extracting the first join key of the first data and an address of the first data from the respective recording means, sorting the extracted first join keys and the addresses so as to produce a first sub-table, and transferring the first sub-table to the main processor;

in the main processor, receiving the first subtables from each of the plurality of sub-processors, merge-sorting the first sub-tables with the first join key so as to produce a first main table;

in each of the plurality of sub-processors, retrieving the second data from the respective recording means, sorting the second data with the second join key, comparing the second join key of the second data with the first join keys included in the first main table, selecting second data whose second join key is matched with one of the first join keys in the first main table, so as to produce a second sub-table, and transferring the second sub-table to the main processor;

in the main processor, receiving the second subtables from each of the plurality of sub-processors, merge-sorting the second sub-tables with the second join key so as to produce a second main table, joining the first and second data based on the first and second main tables and producing the joined table.

12. The join processing method of claim **11**, further comprising the steps of:

in the main processor, producing a first flag field corresponding to each of the first data in the first main table, in each of the plurality of sub-processors, storing a comparison result in the first flag field when the sub-processor compares the second join key of the second data with the first join keys included in the first main table to select the second data;

in the main processor, joining the first and second data according to the comparison result stored in the first flag field in the first main table.

13. A join processing method for joining distributed first and second data, the first data including a first join key, the second data including a second join key and producing a joined table, for a database system including a plurality of recording means for storing the distributed first and second data, a plurality of sub-processors each coupled to one of the plurality of recording means respectively, and a main processor coupled to the plurality of sub-processors for controlling the plurality of sub-processors and producing the joined table, the join processing method comprising the steps of:

in each of the plurality of sub-processors, retrieving the first data with the first join key from the respective recording means, sorting the retrieved first data with the first join key so as to produce a first sub-table, and transferring the first sub-table to the main processor;

in the main processor, receiving the first sub-tables from each of the plurality of sub-processors, merge sorting the first sub-tables with the first join key so as to produce a first main table, calculating first hash values of the first join keys included in the first main table, producing a first hash table showing an existence of the first hash values, and transferring the first hash table to each of the plurality of sub-processors;

in each of the plurality of sub-processors, receiving the first hash table, calculating a second hash value of the second join key of the second data, comparing the second hash value with the first hash values in the first hash table, extracting a second join key and an address of the second data whose second hash value is confirmed to exist in the first hash table, sorting the extracted second join keys and addresses of the second data with the second join key so as to produce a second sub-table, and transferring the second sub-table to the main processor;

in the main processor, receiving the second subtables from each of the plurality of sub-processors, merge sorting the second sub-tables with the second join key so as to produce a second main table, joining the first and second data based on the first and second main tables, and producing the joined table.

14. A join processing method for joining distributed first and second data, the first data including a first join key, the second data including a second join key and producing a joined table, for a database system including a plurality of recording means for storing the distributed first and second data, a plurality of sub-processors each coupled to one of the plurality of recording means respectively, and a main processor coupled to the plurality of sub-processors for controlling the plurality of sub-processors and producing the joined table, the join processing method comprising the steps of:

in each of the plurality of sub-processors, retrieving the first data with the first join key from the respective recording means, sorting the retrieved first data with the first join key so as to produce a first sub-table, and transferring the first sub-table to the main processor;

in the main processor, receiving the first sub-tables from each of the plurality of sub-processors, merge sorting the first sub-tables with the first join key so as to produce a first main table, and transferring the first main table to each of the plurality of sub-processors;

in each of the plurality of sub-processors, receiving the first main table, retrieving the second data in the respective recording means with the second join key equal to a first join key included in the received first main table, joining the first data in the received first main table and the retrieved second data based on the second join key, producing a joined sub-table, and transferring the joined sub-table to the main processor;

in the main processor, receiving the joined sub-tables from each of the plurality of sub-processors, and merging the joined sub-tables with the first and second join key so as to produce the joined table.

15. The join processing method of claim 14 further comprising the steps of:

in each of the plurality of sub-processors, extracting the second join key and an address of the second data, comparing the extracted second join key of the second data with the first join keys included in the first main table, retrieving second data from the recording means based on the extracted address according to a comparison result, and producing the joined sub-table.

16. A join processing method for joining distributed first and second data, the first data including a first join key, the second data including a second join key and producing a joined table, for a database system including a plurality of recording means for storing the distributed first and second data, a plurality of sub-processors each coupled to one of the plurality of recording means respectively, and a main processor coupled to the plurality of sub-processors for controlling the plurality of sub-processors and producing the joined table, the join processing method comprising the steps of:

in each of the plurality of sub-processors, retrieving the first data with the first join key from the respective recording means, sorting the retrieved first data with the first join key so as to produce a first sub-table, dividing the first sub-table into a plurality of first sub-tables corresponding to the respective sub-processors, and transferring the divided first subtables to corresponding sub-processors;

in each of the plurality of sub-processors, receiving the divided first sub-tables from the plurality of sub-processors, merging the divided first sub-tables with the first join key so as to produce a local first main table, retrieving the second data in the recording means with the second join key equal to the first join key included in the local first main table, transferring the retrieved second data to the plurality of subprocessors;

in the each of the plurality of sub-processors, receiving the second data from plurality of the sub-processors, merging the received second data with the second join key so as to produce a local second main table, joining the first data in the received local first main table and the produced second data based on the first and second join key, producing a joined sub-table, and transferring the joined sub-table to the main processor;

in the main processor, receiving the joined sub-tables from each of the plurality of sub-processors, and merging the joined sub-tables with the first and second join key so as to produce the joined table.

* * * * *

28/3,K/32 (Item 32 from file: 350)
DIALOG(R)File 350:Derwent WPIX

011260497 **Image available**
WPI Acc No: 1997-238400/199722
XRPX Acc No: N97-196927
Multiple **bank memory architecture and operation of same - has** first **and** second columns **of memory cells each including conductive bitline plus gate for selectively coupling between them**
Patent Assignee: CIRRUS LOGIC INC (CIRR-N)
Inventor: MOHAN R G R; RAO G R M
Number of Countries: 012 Number of Patents: 007
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 771008	A2	19970502	EP 96307730	A	19961025	199722	B
JP 9231746	A	19970905	JP 96284320	A	19961025	199746	
US 5687132	A	19971111	US 95548752	A	19951026	199751	
KR 97022773	A	19970530	KR 9649887	A	19961026	199823	
KR 258672	B1	20000615	KR 9649887	A	19961026	200131	
EP 771008	B1	20020102	EP 96307730	A	19961025	200205	
DE 69618319	E	20020207	DE 618319	A	19961025	200218	
			EP 96307730	A	19961025		

Priority Applications (No Type Date): US 95548752 A 19951026
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 771008	A2	E	16	G11C-007/00	
Designated States (Regional): BE DE ES FR GB IE IT NL PT					
JP 9231746	A		20	G11C-011/401	
US 5687132	A		13	G11C-008/00	
KR 97022773	A			G06F-013/00	
KR 258672	B1			G11C-011/401	
EP 771008	B1	E		G11C-007/00	
Designated States (Regional): BE DE ES FR GB IE IT NL PT					
DE 69618319	E			G11C-007/00	Based on patent EP 771008

Multiple **bank memory architecture and operation of same...**

...**has** first **and** second columns **of memory cells each including conductive bitline plus gate for selectively coupling between them**

...Abstract (Basic): The memory (20) includes a **first column** of memory cells including at least one conductive bitline (202). A **second column** of memory cells is also present and also includes at least one conductive bitline (202). A gate (203) selectively couples the bitline of the **first column** with the bitline of the **second column** for transferring a bit of data from a selected cell of the **first column** of cells to a selected cell of the **second column** of cells. Preferably, the gate comprises a field effect transistor, and the memory cells comprise dynamic **random access memory** cells...

...ADVANTAGE - Allows for efficient block transfer of data, efficient generation of images on **multiple** displays, together with increased efficiency of display updates...
...Abstract (Equivalent): a **first plurality** of **columns** of memory cells each including at least one conductive bitline...

...a **second plurality** of **columns** of memory cells each including at least one conductive bitline; and...

...a **plurality** of gates organized in independently controlled groups for selectively coupling said bitlines of a selected group of said **first plurality** of **columns** with a group of said bitlines of said **second plurality** of **columns** for transferring a at least one bit of data from a selected cell of said **first plurality** of **columns** of cells to a selected cell of said **second plurality** of **columns** of cells ...

Title Terms: **MULTIPLE** ;

International Patent Class (Main): **G06F-013/00** ...

US005687132A

United States Patent [19]

Rao

[11] Patent Number: 5,687,132

[45] Date of Patent: Nov. 11, 1997

[54] **MULTIPLE-BANK MEMORY ARCHITECTURE AND SYSTEMS AND METHODS USING THE SAME**

[75] Inventor: **G. R. Mohan Rao**, Dallas, Tex.

[73] Assignee: **Cirrus Logic, Inc.**, Fremont, Calif.

[21] Appl. No.: **548,752**

[22] Filed: **Oct. 26, 1995**

[51] **Int. Cl.**[6] **G11C 8/00**

[52] **U.S. Cl.** **365/230.03**; 365/221; 365/230.04

[58] **Field of Search** 365/230.03, 149, 365/221, 230.04

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,893,281	1/1990	Hashimoto	365/230.06
4,987,559	1/1991	Miyauchi et al.	365/189.04
5,121,360	6/1992	West et al.	365/230.03
5,305,284	4/1994	Iwase	365/238.5
5,319,603	6/1994	Watanabe et al.	365/230.04
5,377,154	12/1994	Takasugi	365/221
5,390,139	2/1995	Smith et al.	365/230.03

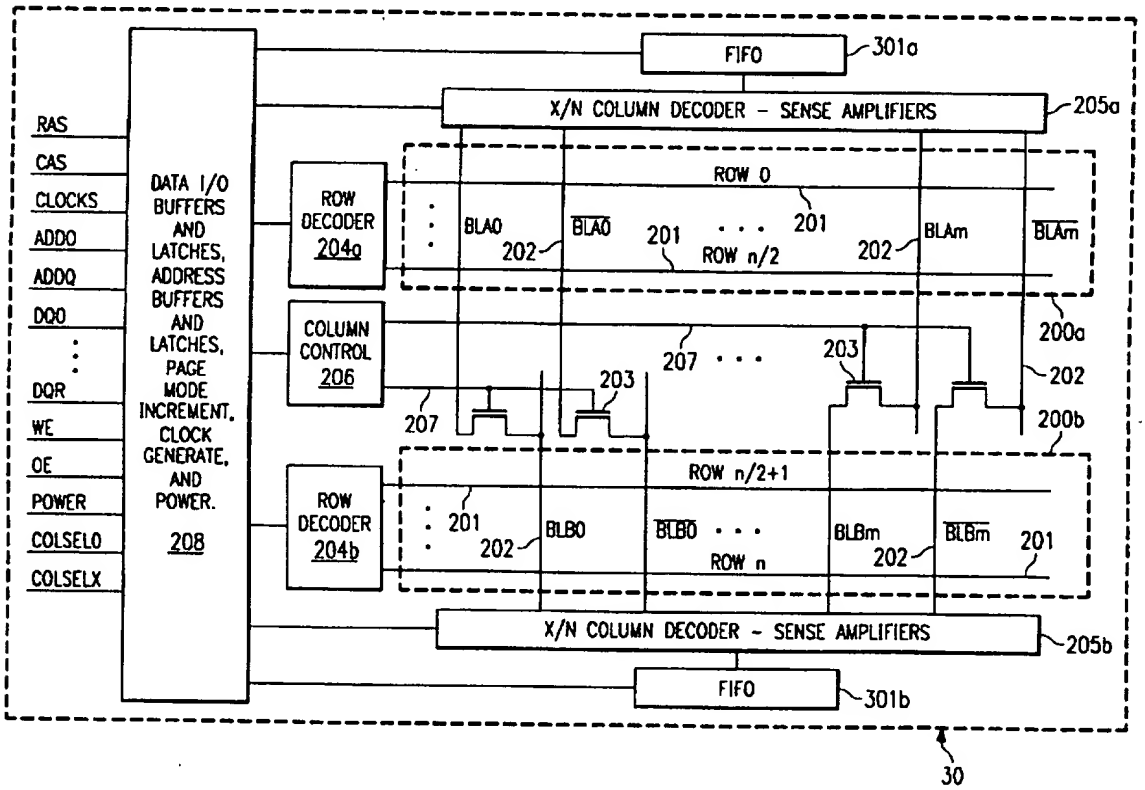
Primary Examiner—Son T. Dinh

Attorney, Agent, or Firm—James J. Murphy

[57] **ABSTRACT**

A memory **20** is disclosed including a first column of memory cells including a conductive bitline **202** and a second column of memory cells also including a conductive bitline **202**. A gate **203** is provided for selectively coupling the bitline **202** of the first column with the bitline **202** of the second column for transferring a bit of data from a selected cell of the first column to a selected cell of the second column.

30 Claims, 4 Drawing Sheets



US-PAT-NO: **5687132**

DOCUMENT-IDENTIFIER: US 5687132 A

TITLE: Multiple-bank memory architecture and systems and methods using the same

DATE-ISSUED: November 11, 1997

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Rao; G. R. Mohan	Dallas	TX	N/A	N/A

US-CL-CURRENT: 365/230.03, 365/221 , 365/230.04

CLAIMS:

What is claimed is:

1. A memory comprising:

a first plurality of **columns** of memory cells each including at least one conductive bitline;

a second plurality of **columns** of memory cells each including at least one conductive bitline; and

a plurality of gates organized in independently controlled groups for selectively coupling said bitlines of a selected group of said first plurality of **columns** with a group of said bitlines of said second plurality of **columns** for transferring a at least one bit of data from a selected cell of said first plurality of **columns** of cells to a selected cell of said second plurality of **columns** of cells.

2. The memory of claim 1 wherein each said gate comprises a field effect transistor.

3. The memory of claim 1 wherein said memory cells comprises dynamic random access memory cells.

4. The memory of claim 1 wherein said first **column** of cells comprises one of a plurality of **columns** of memory cells forming a first subarray and said second **column** of cells comprises one of a plurality of **columns** of cells forming a second subarray.

5. The memory of claim 4 wherein said first and second subarrays each comprise a plurality of **rows** of cells, each said **row** including a conductive wordline, a wordline of a said **row** of said first subarray controlled by a first **row** decoder and a wordline of a said **row** of said second subarray controlled by a second **row** decoder.

6. A memory subsystem comprising:

a first subarray of memory cells arranged in **rows and columns, each said column** associated with a conductive bitline and each said **row** associated with a conductive wordline;

a second subarray of memory cells arranged in **rows and columns, each said column** associated with a conductive bitline and each said **row** associated with a conductive wordline;

circuitry for independently coupling selected groups of said bitlines of said first subarray with corresponding groups of said bitlines of said second subarray;

a first **column** decoder coupled to said bitlines of said first subarray for randomly accessing selected cells along a selected **row** in said first subarray; and

a second **column** decoder coupled to said bitlines of said second

subarray for
randomly accessing selected cells along a selected **row** in said second subarray.

7. The memory subsystem of claim 6 and further comprising **column** control
circuitry operable to cause said circuitry for gating to coupled selected ones
of said bitlines in response to a received control signal.

8. The memory of claim 6 wherein said bitlines of each of said first and
second subarrays comprise folded bitlines.

9. The memory of claim 6 wherein said circuitry for coupling comprises a
plurality of gates.

10. The memory of claim 9 wherein said plurality of gates comprises a
plurality of field effect transistors.

11. The memory of claim 8 and further comprising control circuitry for
causing said circuitry for coupling to couple a said bitline and a
complementary said bitline of said first subarray with a said bitline and a
complementary said bitline of said second subarray in response to a received
control signal.

12. The memory of claim 6 and further comprising a first **row** decoder
coupled to said wordlines of said first subarray and a second **row** decoder
coupled to said wordlines of said second subarray.

13. The memory of claim 12 wherein said first and second **row** decoders
respond to different address sets respectively.

14. The memory of claim 1 wherein said first and second **column** decoders
respond to different address sets respectively.

15. The memory of claim 14 and further comprising a first
first-in-first-out memory coupled to said first **column** decoder and a second
first-in-first-out memory coupled to said second **column** decoder.

16. A memory device comprising:

a first subarray of **rows and columns** of dynamic random access memory cells,
each said **column** including a bitline and each said **row** including a wordline;

a second subarray of **rows and columns** of dynamic random access memory cells,
each said **column** including a bitline and each said **row** including a wordline;

a first **row** decoder for selecting a said wordline in said first subarray in
response to a first set of **row** addresses;

a second **row** decoder for selecting a said wordline in said second subarray
in response to a second set of **row** addresses;

a first **column** decoder for selecting at least one bitline in said first
subarray for random access in response to a first set of **column** addresses;

a second **column** decoder for selecting at least one bitline in said second
subarray for random access in response to a second set of **column** addresses;

a first plurality of gates for selectively coupling at least one bitline of
a corresponding group of bitlines in said first subarray with at least one
bitline of a corresponding group of bitlines in said second subarray, said
first plurality of gates controlled by a first control signal; and

a second plurality of gates for selectively coupling at least one bitline in
a corresponding group of bitlines in said first subarray with at least one
bitline of a corresponding group of bitlines in said second subarray.

17. The memory device of claim 16 wherein first set of **row** addresses is
equivalent to said second set of **row** addresses.

18. The memory device of claim 16 wherein said first set of **column**

addresses is equivalent to said second set of **column** addresses.

19. The memory device of claim 16 wherein said bitlines of said first and second subarrays comprise folded bitlines arranged in pairs of complementary bitlines, a sense amplifier shared by said complementary bitlines of a selected said pair.

20. The memory device of claim 16 and further comprising latching circuitry associated with said first subarray for temporarily latching data read from at least one cell in said first subarray during a transfer through at least one of said gates to a cell in said second subarray.

21. The memory device of claim 16 wherein said plurality of gates comprises a plurality of field effect transistors each having a current path coupling a said bitline in said first subarray with a said bitline in said second subarray.

22. The memory device of claim 16 wherein said plurality of gates are controlled by at least one control signal received from a source external to said memory device.

23. The memory device of claim 16 and further comprising a first FIFO coupled to said first **column** decoder and a second FIFO coupled to said second **column** decoder.

24. A processing system comprising:

a memory device including:

a first subarray of memory cells arranged in **rows and columns, each said column** associated with a conductive bitline and each said **row** associated with a conductive wordline;

a second subarray of memory cells arranged in **rows and columns, each said**

column associated with a conductive bitline and each said **row** associated with a
conductive wordline;

circuitry for randomly accessing at least one selected cell in said first
subarray;

circuitry for randomly accessing at least one selected cell in said second
subarray;

circuitry for coupling a selected one of said bitlines of said first
subarray with a selected one of said bitlines of said second subarray for
transferring data from an accessed cell of said first subarray to an accessed
cell of said second subarray;

a first display device for displaying data received from said first
subarray; and

a second display device for displaying data received from said second
subarray.

25. The processing system of claim 24 wherein said first and second display
devices operate at different refresh rates.

26. The processing system of claim 24 wherein said circuitry for accessing
a said cell in said first subarray and said circuitry for accessing a said cell
in said second subarray are coupled to a single display controller.

27. The processing system of claim 24 wherein said circuitry for accessing
a said cell in said first subarray and said circuitry for accessing a said cell
in said second subarray are coupled to a single core logic chip set.

28. A method for performing a data transfer in a memory subsystem including
a first subarray of memory cells arranged in **rows and columns, each column**
associated with a conductive bitline and each **row** associated with a conductive

wordline, a second subarray of memory cells arranged in **rows and columns, each column** associated with a conductive bitline and each **row** associated with a conductive wordline, and a plurality gates partitioned into at least two independently controllable groups, each group of gates for coupling selected ones of the bitlines of the first subarray with corresponding ones of the bitlines of said second subarray, the method comprising the steps of:

activating a selected wordline in the first subarray;

sensing data at the bitlines of the first subarray from the cells of the selected **row**;

activating a selected group of the gates to couple the sensed data from ones of the bitlines of the first subarray to selected ones of the bitlines of the second subarray; and

activating a selected wordline of the second subarray to write the data from the first subarray into cells of a selected **row and the columns** associated with the selected bitlines of the second subarray.

29. The method of claim 28 and further comprising the step of latching the data at the bitlines of the first subarray after said step of sensing.

30. The method of claim 28 and further comprising the step of deactivating the selected wordline in the first subarray prior to said step of activating the selected wordline in the second subarray.

28/3,K/41 (Item 41 from file: 350)
DIALOG(R)File 350:Derwent WPIX

010684330 **Image available**
WPI Acc No: 1996-181286/199619
XRPX Acc No: N96-152325
Multiple **data registers and addressing technique for block writing of memory of** RAM **- with addressing of colour registers at column address strobe cycle mode rate for loading of registers and block writing of register contents into main memory array**
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC); IBM CORP (IBMC)
Inventor: BUTLER E; SASKI R A; SASAKI R A
Number of Countries: 005 Number of Patents: 005
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 706163	A2	19960410	EP 95480122	A	19950825	199619	B
JP 8106414	A	19960423	JP 95215706	A	19950824	199626	
US 5577193	A	19961119	US 94314112	A	19940928	199701	
EP 706163	A3	19970409	EP 95480122	A	19950825	199728	
JP 3216974	B2	20011009	JP 95215706	A	19950824	200164	

Priority Applications (No Type Date): US 94314112 A 19940928
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 706163	A2	E	13	G09G-001/16	
Designated States (Regional): DE FR GB					
JP 8106414	A		15	G06F-012/00	
US 5577193	A		13	G06F-012/06	
EP 706163	A3			G09G-001/16	
JP 3216974	B2		14	G06F-012/00	Previous Publ. patent JP 8106414

Multiple **data registers and addressing technique for block writing of memory of** RAM **-**

...Abstract (Basic): address one of the memory arrays using a number (Y) of address bits of an **multiple** (X) bit signal...

...address one addressable memory cell of the second memory array to transfer data from the **second** array to the whole **column** of addressed cells in the **first** array...

...ADVANTAGE - In computer graphics displays enhanced frame buffer construction and storage mechanisms. Efficient writing of **multiple** colours...
...Abstract (Equivalent): **first** address means for **column** addressing in a single column address strobe (CAS) cycle a **plurality** of addressable memory cells of said first array of addressable memory cells using Y address...

...one addressable memory cell of said second array of addressable memory cells directly to said **plurality** of addressable memory cells of said first array of addressable memory cells
Title Terms: **MULTIPLE** ;
International Patent Class (Main): **G06F-012/00** ...

... **G06F-012/06**
International Patent Class (Additional): **G06F-012/02** ...
Manual Codes (EPI/S-X): **T01-H01A** ...

US005577193A

United States Patent [19]

Butler et al.

[11] Patent Number: 5,577,193

[45] Date of Patent: Nov. 19, 1996

[54] MULTIPLE DATA REGISTERS AND ADDRESSING TECHNIQUE THEREFORE FOR BLOCK/FLASH WRITING MAIN MEMORY OF A DRAM/VRAM

[75] Inventors: **Edward Butler**, Jonesville, Vt.; **Ronald A. Sasaki**, San Jose, Calif.

[73] Assignee: **International Business Machines Corporation**, Armonk, N.Y.

[21] Appl. No.: **314,112**

[22] Filed: **Sep. 28, 1994**

[51] **Int. Cl.**[6] **G06F 12/06**

[52] **U.S. Cl.** **395/515**; 395/507; 345/201

[58] **Field of Search** 395/131, 162–166; 365/230.03, 230.06, 189.01, 900, 189.02, 230.02, 189.04, 238.5, 189.05; 345/185, 186, 196, 200, 201, 199, 114

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,635,048	1/1987	Nishi et al.	345/203
4,684,942	8/1987	Nishi et al.	345/199
4,804,948	2/1989	Nishi et al.	345/186
4,823,120	4/1989	Thompson et al.	345/199
4,845,477	4/1989	Shibata et al.	345/22
4,857,899	8/1989	Ishii	345/150
4,893,114	1/1990	Ishii	345/22
4,908,700	3/1990	Ishii et al.	348/587
5,038,300	8/1991	Seiler et al.	395/131
5,045,967	9/1991	Igarashi	395/131
5,208,903	5/1993	Curry	395/131
5,218,671	6/1993	Liao et al.	395/131
5,229,971	7/1993	Kiryn et al.	365/230.03
5,319,606	6/1994	Bowen et al.	365/230.06
5,321,809	6/1994	Aranda	395/164
5,392,241	2/1995	Bulter et al.	365/189.01

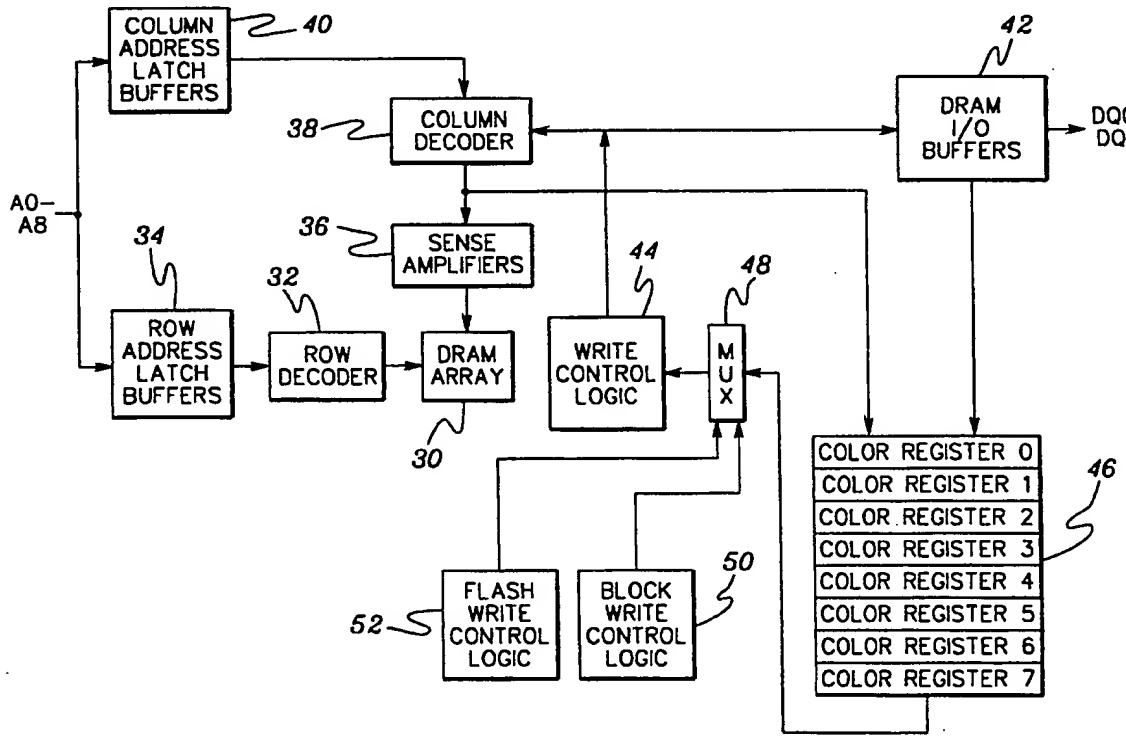
Primary Examiner—Kee M. Tung

Attorney, Agent, or Firm—Heslin & Rothenberg, P.C.

[57] **ABSTRACT**

A frame buffer construction and data storage technique for computer graphics display systems are presented employing a plurality of on-chip color registers. The plurality of on-chip color registers facilitate block writing and flash writing of multiple color information into the main frame buffer video memory. Addressing of the plurality of color registers is achieved at a column address strobe (CAS) cycle page mode rate for both the loading of the registers and the block or flash writing of the contents of the registers into the main memory array.

31 Claims, 6 Drawing Sheets



Loading of data to the multiple registers proceeds as follows. First the "Load Color Reg." signal goes high indicating that one or more of the color registers is to be loaded. Color data is then provided on the VRAM DQ inputs and received at the "Color Data In" pins of the color registers. An address signal is provided on lower order bits CA0, CA1 & CA2 which is decoded to generate the corresponding register address signal CZ0, CZ1, . . . , CZ7. These in turn initiate loading of the color data into the appropriate register through control logic 62 & 64.

Output of data from the color registers is initiated by bringing the "load color reg." signal low. An address signal is presented and decoded such that a register is selected for reading of data therefrom. The selected color register is read, again with only one bit of each color register being depicted in FIG. 6. Once read, the data is output at the "color data" terminal for block or flash writing to the main memory array. Other control logic (not shown) directs the VRAM to look to the color data terminal for the color data to be written into the memory array.

Those skilled in the art will note from the above discussion that an enhanced frame buffer construction and data storage technique for a computer graphics display system are presented herein, wherein a plurality of on-chip color registers facilitate block writing of multiple color information to frame buffer video memory. Significantly increased VRAM performance is obtained because addressing of the plurality of registers is achieved at a CAS cycle time in page mode fashion for both the loading of the multiple on-chip registers and the writing of contents of the registers into the main DRAM array. In the specific embodiment presented, all eight color registers can be loaded in a single RAS cycle and the main array can be written with selected data from the multiple registers in a single RAS cycle. Thus, enhanced performance is obtained where multiple color renderings are desired. Further, depending upon the specific colors to be rendered, loading of the multiple on-chip color registers prior to writing of data to the main memory array may be unnecessary in every instance.

Although specific embodiments of the present invention have been illustrated in the accompanying drawings and described in the foregoing detailed description, it will be understood that the invention is not limited to the particular embodiments described herein, but is capable of numerous re-arrangements, modifications and substitutions without departing from the scope of the invention. For example, although presented herein with respect to writing of "color" data into the memory array, the multiple registers and addressing technique could also be employed for block or flash writing of other types of data into a main memory array. In addition, writing to the main memory array could encompass writing data from one register to a single memory location in the array. The following claims are intended to encompass all such modifications.

We claim:

1. A memory system comprising:

a first memory means having a first array of addressable memory cells for storing data;

a second memory means having a second array of addressable memory cells for storing data, said second array of addressable memory cells being independently addressable from said first array of addressable memory cells, and wherein said second array of addressable memory cells is coupled to said first array of addressable memory cells for transfer of data from said second array of addressable memory cells directly to said first array of addressable memory cells;

first address means for column addressing in a single column address strobe (CAS) cycle a plurality of addressable memory cells of said first array of addressable memory cells using Y address bits of an X address bit signal, wherein X>Y such that unused address bits in said X address bit signal exist; and

second address means for employing said unused address bits of said X address bit signal in said single CAS cycle to address one addressable memory cell of said second array of addressable memory cells for transfer of data from said one addressable memory cell of said second array of addressable memory cells directly to said plurality of addressable memory cells of said first array of addressable memory cells.

2. The memory system of claim 1, wherein said second array of addressable memory cells of said second memory means comprises an array of multi-bit registers such that said one addressable memory cell of said second array of addressable memory cells comprises one multi-bit register.

3. The memory system of claim 2, wherein said array of multi-bit registers comprises an array of multi-bit color registers such that said one multi-bit register stores color data for transfer to said plurality of addressable memory cells of said first array of addressable memory cells.

4. The memory system of claim 3, wherein said memory system comprises a video random access memory chip.

5. The memory system of claim 1, further comprising writing means for block writing data from the one addressable memory cell of the second array of addressable memory cells directly to said plurality of addressable memory cells of the first array of addressable memory cells.

6. The memory system of claim 5, wherein said second array of addressable memory cells stores color data such that said writing means block writes color data from said one addressable memory cell of the second array of addressable memory cells to said plurality of addressable memory cells of the first array of addressable memory cells.

7. The memory system of claim 6, wherein said second array of addressable memory cells comprises an array of eight color registers and wherein said second address means includes means for employing said unused address bits of said X address bit signal in said single CAS cycle to address one color register of said array of eight color registers.

8. The memory system of claim 7, wherein said writing means includes means for block writing color data from said array of eight color registers to said first array of addressable memory cells in a page mode fashion.

9. The memory system of claim 5, wherein said writing means includes means for block writing data from multiple address memory cells of the second array of addressable memory cells directly to the first array of addressable memory cells in a page mode fashion.

10. The memory system of claim 1, wherein said first array of addressable memory cells comprises a dynamic random access memory array.

11. The memory system of claim 1, further comprising means for loading data into the second array of addressable memory cells in a page mode fashion.

12. A memory system comprising:

a memory array having a plurality of addressable memory locations;

a plurality of multi-bit registers for storing data, each multi-bit register of said plurality of multi-bit registers being separately addressable; and

addressing means for selecting one multi-bit register of said plurality of multi-bit registers and an addressable memory location of said plurality of addressable

memory locations in said memory array for transfer of data from said one multi-bit register to said addressable memory location in response to a single, multi-bit address signal.

13. The memory system of claim 12, further comprising means for transferring data from said one multi-bit register to said addressable memory location in a single CAS cycle in response to the single, multi-bit address signal.

14. The memory system of claim 12, wherein said plurality of multi-bit registers comprises a plurality of multi-bit color registers for storing color data.

15. The memory system of claim 12, wherein said memory system comprises a video random access memory chip.

16. The memory system of claim 15, wherein said memory array comprises a dynamic random access memory array.

17. The memory system of claim 12, further comprising means for loading data into said plurality of multi-bit registers in a page mode fashion.

18. The memory system of claim 12, further comprising means for block writing data from said one multi-bit register of said plurality of multi-bit registers to multiple addressable memory locations of said plurality of addressable memory locations in response to the single, multi-bit address signal.

19. The memory system of claim 12, further comprising means for flash writing data from said one multi-bit register of said plurality of multi-bit column registers to multiple addressable memory locations of said plurality of addressable memory locations in response to said single, multi-bit address signal.

20. A graphics system comprising:

a display device having an array of pixels upon which an object is to rendered;

a graphics processor for generating color and coordinate data for each pixel of the object to be rendered;

a raster processor coupled to the graphics processor and to the display device, said raster processor having multiple memory systems, each memory system including

a main memory array having a plurality of addressable memory locations,

a plurality of multi-bit registers for storing color data, each multi-bit register of said plurality of multi-bit registers being separately addressable, and

addressing means for selecting one multi-bit register of said plurality of multi-bit registers and multiple addressable memory locations of said plurality of addressable memory locations in said main memory array for transfer of color data from said one multi-bit register to said multiple addressable memory locations in response to a single, multi-bit address signal; and

means for transferring data stored in said memory systems of said raster processor to the array of pixels of said display device for rendering of said object.

21. The graphics system of claim 20, wherein each memory system of said multiple memory systems comprises a dynamic random access memory (DRAM) or a video random access memory (VRAM).

22. The graphics system of claim 20, wherein each memory system includes means for block writing data from said plurality of multi-bit registers to said memory array in a page mode fashion.

23. The graphics system of claim 20, wherein each memory system of said multiple memory systems includes means for loading data into said plurality of multi-bit registers in a page mode fashion.

24. The graphics system of claim 20, wherein said plurality of multi-bit registers comprises eight multi-bit registers, each multi-bit register of said eight multi-bit registers being capable of storing a different color data.

25. A method for block writing in a single row address strobe (RAS) cycle different color data to different blocks of memory cells of a main memory array using a plurality of associated color registers, each color register being separately addressable and at least some of said color registers storing different color data, said method comprising the steps of:

(a) initiating a RAS cycle;

(b) decoding a first column address signal to select one color register of said plurality of associated color registers and to select multiple memory cells of said main memory array;

(c) block writing color data from said one color register to said multiple memory cells of said main memory array in said step (c);

(d) accomplishing said steps (b) & (c) within a first single CAS cycle; and

(e) repeating said steps (b) & (c) in a second single CAS cycle for a second column address signal such that different color data is written to different multiple memory cells of the main memory array within said RAS cycle.

26. The method of claim 25, further comprising the steps of:

successively repeating said steps (b) & (c) in additional single CAS cycles for additional column address signals such that multiple different color data is written to different multiple memory cells of the main memory array within said RAS cycle; and

terminating said RAS cycle.

27. A method for loading color data into multiple registers of a video memory system having a main memory array of addressable memory cells and multiple separately addressable registers coupled thereto, said method comprising the steps of:

(a) monitoring when a load color register signal transitions active;

(b) receiving color data at a data input to the video memory system;

(c) during a CAS cycle, receiving and decoding predefined address bits of an input column address signal to identify one register of said multiple registers for storing the color data of said step (b); and

(d) during said CAS cycle, loading said color data into the one register identified in said step (c).

28. The method of claim 27, further comprising repeating said steps (b)–(d) in sequential CAS cycles such that color data is loaded into said multiple registers in a page mode fashion.

29. The method of claim 27, wherein said multiple registers comprise eight multi-bit color registers and wherein said receiving step (c) includes decoding three predefined address bits of the input column address signal to identify one register of said eight multi-bit color registers for storing of the color data.

30. In a memory system having a main memory array of addressable memory cells and multiple data registers independently addressable from the main memory array and coupled to the main memory array for transfer of data thereto, a method for block writing data from one register of said multiple data registers to a plurality of addressable

memory cells of said main memory array, said method comprising the steps of:

(a) receiving and decoding a single, multi-bit address signal, said single, multi-bit address signal identifying one data register of said multiple data registers and a partial address of a plurality of addressable memory locations of said main memory array; and

(b) block writing data from the one data register of said multiple data registers to said plurality of addressable memory locations of said main memory array.

31. The method of claim 30, further comprising repeating said steps (a) & (b) for multiple multi-bit address signals such that data from said multiple data registers is block written to the main memory array in a page mode fashion.

* * * * *

28/3,K/35 (Item 35 from file: 350)
DIALOG(R)File 350:Derwent WPIX

010998001 **Image available**
WPI Acc No: 1996-494950/199649
XRPX Acc No: N96-417464
Non - volatile **solid state** memory **unit with floating gate type transistor e.g.** EEPROM **- in which bit line linked to first page buffer and to second page buffer are arranged alternately**
Patent Assignee: SAMSUNG ELECTRONICS CO LTD (SMSU)
Inventor: CHOE B; LIM Y; CHOI B
Number of Countries: 003 Number of Patents: 003
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 8255496	A	19961001	JP 95329281	A	19951218	199649	B
US 5625590	A	19970429	US 95574970	A	19951219	199723	
KR 140179	B1	19980715	KR 9435016	A	19941219	200018	

Priority Applications (No Type Date): KR 9435016 A 19941219
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 8255496	A		20	G11C-016/06	
US 5625590	A		27	G11C-007/00	
KR 140179	B1			G11C-016/06	

Non - volatile **solid state** memory **unit with floating gate type transistor e.g.** EEPROM -

...Abstract (Basic): The memory unit includes **multiple** floating gate type memory cells (M1-M16) which are arranged in line sequences in shape of a matrix. **Multiple** word lines (WL0-WL15) that are linked to the memory cell are arranged in the...

... **Multiple** bit lines (BL0-BL4.094) arranged parallelly are connected to the memory cells arranged in...

...out operation by using two page buffers. Shortens data loading cycle time. Provides high density **integration** by narrowing bit line pitch
...Abstract (Equivalent): A **nonvolatile** semiconductor memory comprising ...

...a **plurality** of floating gate type memory cells arranged in rows and columns...

...a **plurality** of word lines, each of said word lines being connected to the memory cells arranged...
...a **plurality** of bit lines arranged in parallel to one another, each of said bit lines being connected to the memory cells arranged in a given one of said columns, said **plurality** of bit lines being divided into a **plurality** of bit line groups, each having a pair of adjacent bit lines; and...

...circuit including a first page buffer portion connected to bit line groups extending in a **first column** direction and a **second** page buffer portion connected to bit line groups extending to a **second column** direction, each of said **plurality** of bit line groups alternately extending to either one of said **first** and **second column** directions, said **first column** direction being opposite to said **second column** direction...
...Title Terms: **EEPROM** ;

US005625590A

United States Patent [19]

Choi et al.

[11] Patent Number: 5,625,590

[45] Date of Patent: Apr. 29, 1997

[54] **NONVOLATILE SEMICONDUCTOR MEMORY**

[75] Inventors: **Beyng-Sun Choi; Young-Ho Lim**, both of Suwon, Rep. of Korea

[73] Assignee: **Samsung Electronics Co., Ltd.**, Suwon, Rep. of Korea

[21] Appl. No.: **574,970**

[22] Filed: **Dec. 19, 1995**

[30] **Foreign Application Priority Data**

Dec. 19, 1994 [KR] Rep. of Korea 35016/1994

[51] **Int. Cl.**[6] **G11C 7/00**

[52] **U.S. Cl.** **365/185.17**; 365/185.11; 365/230.04; 365/189.05

[58] **Field of Search** 365/185.17, 185.11, 365/185.13, 230.03, 230.04, 230.05, 189.05, 230.08

[56] **References Cited**

U.S. PATENT DOCUMENTS

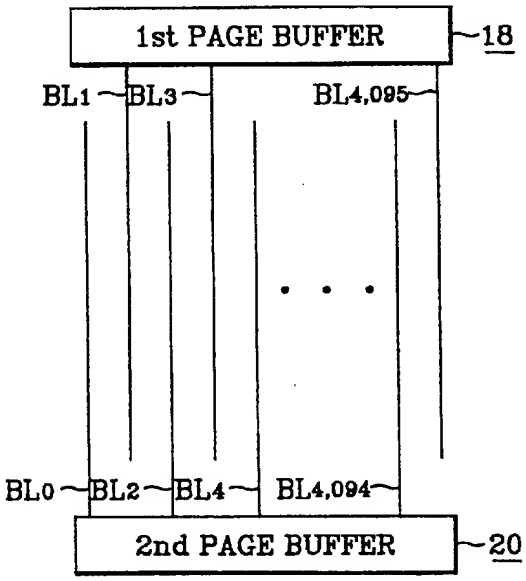
4,825,098	4/1989	Aoyama	365/230.05
4,961,171	10/1990	Pinkham et al.	365/230.05
5,416,742	5/1995	Takada	365/230.04

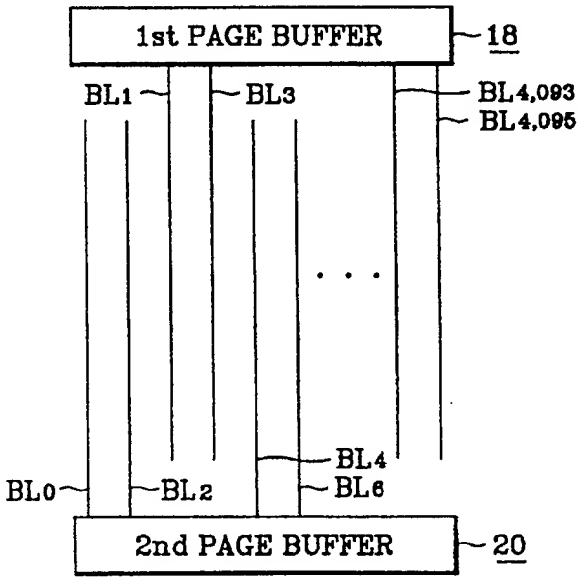
Primary Examiner—David C. Nelms
Assistant Examiner—Vu A. Le
Attorney, Agent, or Firm—Cushman Darby & Cushman, IP Group of Pillsbury Madison & Sutro LLP

[57] **ABSTRACT**

A nonvolatile semiconductor memory according to the present invention includes a plurality of floating gate type memory cells arranged in rows and columns, a plurality of word lines, each word line being connected to the memory cells arranged in a given one of the rows, and a plurality of bit lines arranged in parallel to one another. Each bit line is connected to the memory cells arranged in a given one of the columns and the bit lines are divided into first bit lines and second bit lines which are arranged to alternate with one another. A page buffer is also included which has a first page buffer portion connected to the first bit lines extending in a first column direction and a second page buffer portion connected to the second bit lines extending in a second column direction, the first column direction being opposite to the second column direction. This construction allows for high integration density and reduced data loading cycle time.

4 Claims, 16 Drawing Sheets





art without departing from the scope and spirit of this invention. Accordingly, it is not intended that the scope of the claims appended hereto be limited to the description as set forth herein, but rather that the claims be construed as encompassing all the features of patentable novelty that reside in the present invention, including all features that would be treated as equivalents thereof by those skilled in the art which this invention pertains.

What is claimed is:

1. A nonvolatile semiconductor memory comprising:

a plurality of floating gate type memory cells arranged in rows and columns;

a plurality of word lines, each of said word lines being connected in common to the memory cells arranged in a given one of said rows;

a plurality of bit lines arranged in parallel to one another, each of said bit lines being connected to the memory cells arranged in a given one of said columns, said bit lines being divided into alternatingly arranged first bit lines and second bit lines; and

a page buffer circuit including a first page buffer portion connected to said first bit lines extending in a first column direction and a second page buffer portion connected to said second bit lines extending in a second column direction, said first column direction being opposite to said second column direction.

2. The nonvolatile semiconductor memory as set forth in claim 1, further comprising a first row decoder connected to the word lines extending in a first row direction and a second row decoder connected to the word lines extending in a second row direction, said plurality of word lines being arranged in parallel one another, each of said word lines alternately extending in one of said first and second row directions, said first row direction being opposite to said second row direction.

3. The nonvolatile semiconductor memory as set forth in claim 1, wherein each of said first and second page buffer portions comprise a data latch and sensing circuit connected to a corresponding bit line.

4. A nonvolatile semiconductor memory comprising:

a plurality of floating gate type memory cells arranged in rows and columns;

a plurality of word lines, each of said word lines being connected to the memory cells arranged in a given one of said rows;

a plurality of bit lines arranged in parallel to one another, each of said bit lines being connected to the memory cells arranged in a given one of said columns, said plurality of bit lines being divided into a plurality of bit line groups, each having a pair of adjacent bit lines; and

a page buffer circuit including a first page buffer portion connected to bit line groups extending in a first column direction and a second page buffer portion connected to bit line groups extending to a second column direction, each of said plurality of bit line groups alternately extending to either one of said first and second column directions, said first column direction being opposite to said second column direction.

* * * * *

28/3,K/46 (Item 46 from file: 350)
DIALOG(R)File 350:Derwent WPIX

THE APPLICANT

010475558 **Image available**
WPI Acc No: 1995-376879/199549

DRAM **controller for computer system - activates RAS when selected low address is same as low address stored and deactivates RAS when addresses are not same**

Patent Assignee: NEC CORP (NIDE)
Inventor: KUSUDA M
Number of Countries: 002 Number of Patents: 002
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 7248963	A	19950926	JP 9462139	A	19940308	199549	B
US 5644747	A	19970701	US 95398442	A	19950303	199732	

Priority Applications (No Type Date): JP 9462139 A 19940308
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 7248963	A		23		
US 5644747	A		17		

DRAM **controller for computer system**...

...Abstract (Basic): The **DRAM** controller has a decoder. When the first **DRAM** group is accessed in high speed mode, the RAS signal is made active. A comparator...

...two are same, then the RAS is kept active. When the access of the first **DRAM** group is terminated...

...address, then the RAS is deactivated. Subsequently while passing a new low address for accessing **DRAM** , the RAS is activated again...

...ADVANTAGE - Shortens mean access time since low address value of **DRAM** is maintained. Increases working speed by employing address comparator. Does not increases hardware scale...

...Abstract (Equivalent): A memory controller for a computer system having a processor and a **plurality** of groups of memories each being capable of buffering a row address, wherein one group...

...a **plurality** of latch means corresponding respectively to said groups of memories, each of the latch means...

...by said comparator for directing access to the group of memories currently selected using the **first row** address...

...said control means being responsive to a subsequent coincidence detected by said comparator when the **first row** address is stored in the selected latch means for directing access to the memories of the currently selected group using the same **first row** address and one of a **plurality** of column addresses from said processor, and...

...said control means being responsive to a noncoincidence detected by said comparator when the **first row** address is stored in the selected latch means for directing access to the memories of the currently selected group using a **second row** address and one of said column addresses from said processor...

Title Terms: **DRAM** ;

International Patent Class (Main): **G06F-012/00** ...

... **G06F-012/02**
Manual Codes (EPI/S-X): **T01-H01A**

US005644747A

United States Patent [19]

Kusuda

[11] Patent Number: 5,644,747

[45] Date of Patent: Jul. 1, 1997

[54] **PAGE MODE ACCESS MEMORY CONTROLLER INCLUDING PLURAL ADDRESS LATCHES AND A COMPARATOR**

[75] Inventor: **Masahiro Kusuda**, Tokyo, Japan

[73] Assignee: **NEC Corporation**, Tokyo, Japan

[21] Appl. No.: **398,442**

[22] Filed: **Mar. 3, 1995**

[30] **Foreign Application Priority Data**

Mar. 8, 1994 [JP] Japan 6-062139

[51] **Int. Cl.**[6] **G06F 12/00**; G11C 8/00

[52] **U.S. Cl.** **395/405**; 395/421.07; 395/421.01; 395/419; 395/494; 365/230.08; 365/230.02; 365/189.07

[58] **Field of Search** 395/405, 481, 395/494, 419, 421.01, 421.07, 421.08; 365/233, 193, 230.02, 230.08, 238.5, 189.07

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,870,622	9/1989	Aria et al.	365/238.5
4,933,910	6/1990	Olson et al.	365/233
5,034,917	7/1991	Bland et al.	395/494
5,269,010	12/1993	MacDonald	395/405
5,274,786	12/1993	Diehl	395/421.08
5,303,364	4/1994	Mayer et al.	395/467
5,305,277	4/1994	Derwin et al.	365/230.08
5,436,869	7/1995	Yoshida	365/230.08
5,522,064	5/1996	Aldereguia et al.	395/494

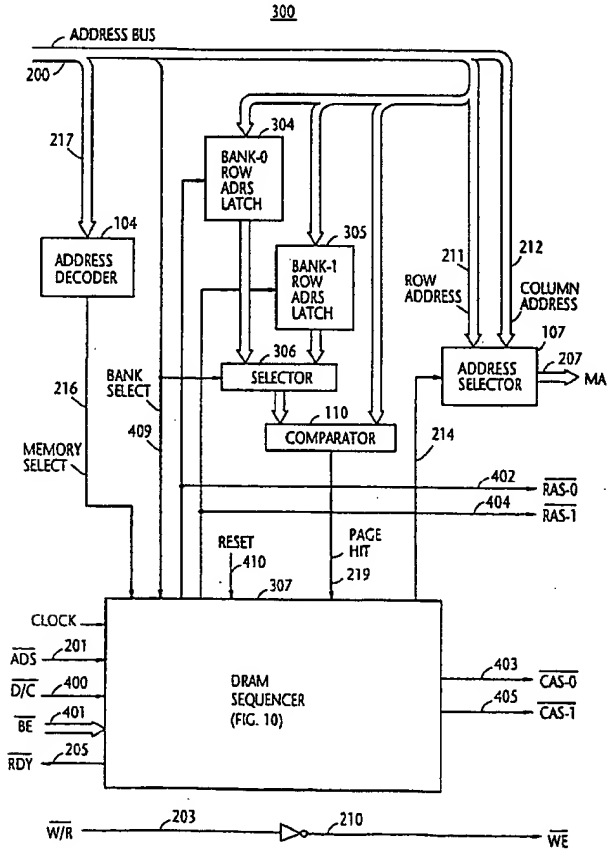
Primary Examiner—Glenn Gossage

Attorney, Agent, or Firm—Whitham, Curtis, Whitham & McGinn

[57] **ABSTRACT**

A memory controller for a computer system having a processor and dynamic random access memories (DRAMs) which are grouped into banks, and having multiple latches associated respectively with the groups of DRAMs. Each latch stores a row address from the processor in response to a row address strobe (RAS) signal when the corresponding memory bank is selected and one of the latches is selected corresponding to the selected memory bank. The row address stored in the selected latch is compared by a comparator with a row address from the processor. Initially, a controller accesses the DRAMs of a bank currently selected using a first row address from the processor. If the same row of the selected bank is addressed again, a coincidence is detected by the comparator, and in response, the controller accesses the DRAMs using the same row address and a column address from the processor. If a noncoincidence is detected by the comparator, the controller accesses the DRAMs of the currently selected bank using a second row address and a column address from the processor.

8 Claims, 11 Drawing Sheets



mitted to the row address strobe line **404** following a delay introduced by delay element **335**, so that the row address strobe is supplied to the bank-1 memories **303** after the setup time of the module. The bank-1 row address latch **305** is thus enabled to store a row address code "**200**" from microprocessor **100**. This address code is supplied through the page address selector **306** to the row address comparator **110**. During the second half section of period 5, a match occurs between the stored row address and the one supplied on the row address bus **211**, and comparator **110** activates a high level on the page hit line **219**.

With the row address strobe line **404** being set to low, the AND gate **337** produces a low-level output, which causes flip-flop **316** to change to a low output state, thus activating a low level on the MA select line **214** at the start timing of period 6. A column address code "010" on bus **212** is selected by selector **107** and supplied to the bank-1 module **303** through the MA output line **207**. With the output of flip-flop **333** being set during period 5, flip-flop **340** switches to a high output state at the start of period 6. As a result, AND gate **336** produces a high-level output which is applied through delay element **338** to NAND gate **339**, so that it supplies a signal to the NAND-gate bank **320** to pass a byte enable (BE) pulse from microprocessor **100** on the $\overline{\text{CAS-1}}$ line **405** to the bank-1 memories **303**, following the setup time of the bank-1 memories. With the output of NAND gate **339** being set to low, the ready line $\overline{\text{RDY}}$ **205** is made active low during the second half section of period 6, informing the microprocessor **100** of the end of the second RAS-CAS bus cycle. As a result, a data read operation from the address "00600040" of the bank-1 memories is terminated and a row address code "200" is stored in the bank-1 latch **305** during period 6.

With the row address codes "000" and "200" being stored in the bank-0 latch **304** and the bank-1 latch **305**, respectively, if the microprocessor **100** attempts a third data read operation during period 7 by accessing an address "00000004" of the bank-0 memories using the row address code "000" and a column address code "001", the page hit line **219** is quickly set to high level immediately following the low-setting of the address strobe line **201**. Flip-flop **311** is switched to high level at the start of period 8. With the bank select line **409** being set to low, AND gate **326** produces an output, causing the $\overline{\text{CAS-0}}$ line **403** to produce a active low during period 8. With the MA select line **214** still being set to low, the column address "001" is applied to the bank-0 memories during period 8. In a similar manner, if the microprocessor **100** attempts a fourth data read operation during period 9 by accessing an address "00600044" of the bank-1 memories using the row address code "200" and a column address code "011", the page hit line **219** is quickly set to high level immediately following the low-setting of the address strobe line **201** by microprocessor **100**. Flip-flop **311** is switched to high output state at the start of period 10. With the bank select line **409** being set to high, AND gate **336** produces an output, causing the $\overline{\text{CAS-1}}$ line **405** to produce a active low pulse during period 10. With the MA select line **214** being still set to low, the column address "011" is sent to the bank-1 memories during period 10.

With the row address codes "000" and "200" still being stored in the bank-0 latch **304** and the bank-1 latch **305**, respectively, assume that the microprocessor **100** attempts a fifth read operation during period 11 by accessing an address "0001000C" of the bank-0 memories using the row address code "010" and a column address code "003". A page miss occurs immediately following the low-level setting of the address strobe line **201**, setting the page hit line **219** to low. Thus, a new bus cycle begins by setting the AND gate **310** to high, and AND gate **321** is conditioned to reset the flip-flop **323** at the start of period 12 and the $\overline{\text{RAS-0}}$ line **402** is set to inactive high state. At the start of period 13, flip-flop **316** is switched to high level for coupling the new row address code "010" to the bank-0 memories through the MA output line **207**, while storing it into the bank-0 row address latch **304**. As a result, a page hit occurs at the midpoint of period 13, setting the page hit line **219** to high to cause the $\overline{\text{CAS-0}}$ line **403** to produce a low-level pulse during period 14.

If the microprocessor **100** accesses an address "0060048" of the bank-1 memories during period 15, using the row address code "200" and a column address code "012", a page hit occurs quickly and the $\overline{\text{CAS-1}}$ line **405** is set to the active low state during period 16. In like manner, if the microprocessor **100** accesses an address "00010010" of the bank-0 memories during period 17, using the row address code "010" and a column address code "004", a page hit occurs quickly and the $\overline{\text{CAS-0}}$ line **403** is set to the active low state during period 18.

It is seen from the foregoing description that since active row address values are stored respectively for different banks of memories, there is a less likelihood of the occurrences of a page miss even if access to the memories changes frequently from one bank to another, so that the average access time can be reduced.

What is claimed is:

1. A memory controller for a computer system having a processor and a plurality of groups of memories each being capable of buffering a row address, wherein one group of memories of said groups of memories is selected by said processor, said memory controller comprising:

a plurality of latch means corresponding respectively to said groups of memories, each of the latch means storing a row address from said processor when the corresponding group of memories is selected by said processor;

selector means for selecting one of said latch means corresponding to the group of memories which is currently selected by said processor;

a comparator for comparing the row address stored in the selected latch means with a row address from said processor to detect a coincidence or a noncoincidence therebetween; and

control means responsive to an initial coincidence detected by said comparator for directing access to the group of memories currently selected using the first row address,

said control means being responsive to a subsequent coincidence detected by said comparator when the first row address is stored in the selected latch means for directing access to the memories of the currently selected group using the same first row address and one of a plurality of column addresses from said processor, and

said control means being responsive to a noncoincidence detected by said comparator when the first row address is stored in the selected latch means for directing access to the memories of the currently selected group using a second row address and one of said column addresses from said processor.

2. A memory controller as in claim **1**, wherein said control means includes an address selector for outputting an address signal to said groups of memories.

3. A memory controller as in claim **2**, wherein said control means comprises a plurality of memory bank controllers,

each connected to an OR gate, and a flip-flop connected to said OR gate, wherein said flip-flop outputs a control signal for controlling said address selector.

4. A memory controller as in claim 3, wherein each of said memory bank controllers outputs a row address strobe (RAS) signal to a corresponding group of memories of said group of memories.

5. A memory controller for a computer system having a processor and a plurality of groups of dynamic random access memories (DRAMs), wherein one group of DRAMs of said groups of DRAMs is selected by said processor, and the selected group of DRAMs is row-accessed by activating a row address strobe (RAS) line of that group and each memory of the row-accessed group is column-accessed by activating a column address strobe (CAS) line of that memory, said memory controller comprising:

a plurality of latch means corresponding respectively to said groups of DRAMs, each of the latch means storing a row address from said processor when the corresponding group of DRAMs is selected by said processor;

selector means for selecting one of said latch means corresponding to the group of DRAMs which is currently selected by said processor;

a comparator for comparing the row address stored in the selected latch means with a row address from said processor to detect a coincidence or a noncoincidence therebetween; and

control means responsive to an initial coincidence detected by said comparator when the first row address is stored in the selected latch means for activating a RAS line of the currently selected group of DRAMs and a CAS line of one of the DRAMs of the currently selected group and for supplying the first row address and one of a plurality of column addresses to said DRAMs,

said control means being responsive to a subsequent coincidence detected by said comparator when the first row address is stored in the selected latch means for activating the CAS line of one of the DRAMs of the currently selected group and supplying one of said column addresses from said processor to said DRAMs, and directing access to the memories of the currently selected group using the same first row address and one of a plurality of column addresses from said processor, and

said control means being responsive to a noncoincidence detected by said comparator when the first row address is stored in the selected latch means for inactivating and then activating the RAS line of the currently selected group, activating the CAS line of one of the DRAMs of the currently selected group, and supplying a second row address and one of said column addresses from said processor to said DRAMs.

6. A memory controller as in claim 5, wherein said control means includes an address selector for outputting an address signal to said groups of DRAMs.

7. A memory controller as in claim 6, wherein said control means comprises a plurality of memory bank controllers, each connected to an OR gate, and a flip-flop connected to said OR gate, wherein said flip-flop outputs a control signal for controlling said address selector.

8. A memory controller as in claim 8, wherein each of said memory bank outputs a row address strobe (RAS) signal to a corresponding group of DRAMs of said group of DRAMs.

* * * * *

```
Set     Items   Description
S1    2066519   COMBINE? OR COMBINING? OR MERGE? OR MERGING? OR AMALGAM?
S2     220734   BLEND? OR (ADD OR ADDS OR ADDING OR ADDED OR ADDITION)()TO-
              GETHER?
S3      41683   COMMINGL? OR CONJOIN? OR MINGL? OR INTERMIX? OR INTERMINGL?
              OR COMMIX? OR BIND?()TOGETHER?
S4    5425634   FUSE? OR FUSING? OR MELD? OR UNITE? OR UNITING? OR INTERBL-
              END? OR INTEGRAT?
S5    1921765   INTERWEAV? OR INTERWOVEN? OR HOMOG? OR JOIN?
S6    9148347   FIRST? OR 1ST OR PRIMARY OR INITIAL? OR ORIGINAL? OR LEADO-
              FF? OR MAIN OR CHIEF OR INTRODUCTORY?
S7    7488971   SECOND? OR 2ND OR DOUBL? OR TWIN? OR EXTRA? OR DUPLICAT? OR
              ANOTHER OR SUBSIDIAR? OR AUXILIAR?
S8      10837   STORAG?()MODUL?
S9       5563   VOLATIL?(3N)(MEMOR? OR STORAG? OR CACHE? OR BUFFER?)
S10     274343   RAM? ? OR SRAM? ? OR DRAM? ? OR SDRAM? ? OR VRAM? ? OR RAN-
              DOM()ACCESS()MEMOR? OR CDROM? OR CD()ROM? ?
S11     212692   ROM? ? OR PROM? ? OR EPROM? ? OR EEPROM? ? OR NONVOLATIL? -
              OR NON()VOLATIL? OR NVROM? OR READ()ONLY()MEMOR?
S12      85068   HD OR HDS OR HARDDRIVE? OR HARDDISC? OR HARDDISK? OR HARD(-
              )(DRIVE? OR DISC? OR DISK?) OR DISCDRIVE? OR DISKDRIVE?
S13      38644   (DISC OR DISK)()DRIVE? OR STOR?()(DISK? OR DISC?) OR HDD OR
              HDDS OR CDRW OR CDW OR CD()(W OR RW)
S14     323155   ROW? ?
S15     611809   TABLE? ?
S16     796751   COLUMN? ?
S17    7123990   PLURAL? OR MULTIP? OR MULTIT? OR SEVERAL? OR NUMEROUS? OR -
              MANY
S18         10   S1:S5 AND S8:S13 AND S14 AND S15 AND S16
S19         41   S1:S5 AND S8:S13 AND S14:S16(5N)S6:S7 AND S17
S20        458   S1:S5 AND (S6 AND S7 AND S17)(7N)S8:S13
S21         22   S20 AND S14:S16
S22         67   S18:S19 OR S21
S23         51   S22 AND PY<2002
S24         46   RD  (unique items)
File   2:INSPEC 1898-2005/Dec W2
         (c) 2005 Institution of Electrical Engineers
File   6:NTIS 1964-2005/Dec W2
         (c) 2005 NTIS, Intl Cpyrght All Rights Res
File   8:Ei Compendex(R) 1970-2005/Dec W2
         (c) 2005 Elsevier Eng.  Info. Inc.
File  34:SciSearch(R) Cited Ref Sci 1990-2005/Dec W2
         (c) 2005 Inst for Sci Info
File  35:Dissertation Abs Online 1861-2005/Nov
         (c) 2005 ProQuest Info&Learning
File  65:Inside Conferences 1993-2005/Dec W3
         (c) 2005 BLDSC all rts. reserv.
File  94:JICST-EPlus 1985-2005/Oct W3
         (c)2005 Japan Science and Tech Corp(JST)
File  99:Wilson Appl. Sci & Tech Abs 1983-2005/Oct
         (c) 2005 The HW Wilson Co.
File 111:TGG Natl.Newspaper Index(SM) 1979-2005/Dec 16
         (c) 2005 The Gale Group
File 144:Pascal 1973-2005/Dec W2
         (c) 2005 INIST/CNRS
File 239:Mathsci 1940-2005/Jan
         (c) 2005 American Mathematical Society
File 256:TecInfoSource 82-2005/Feb
         (c) 2005 Info.Sources Inc
```